N89-25205

MULTILEVEL DECOMPOSITION of COMPLETE VEHICLE CONFIGURATION in a PARALLEL COMPUTING ENVIRONMENT

VINAY BHATT
and
K. M. RAGSDELL
Design Productivity Center
University of Missouri
Columbia, MO

PRECEDING PAGE BLANK NOT FILMED

ABSTRACT

This research summarizes various approaches to multilevel decomposition to solve large structural problems. A linear decomposition scheme based on the Sobieski algorithm is selected as a vehicle for automated synthesis of a complete vehicle configuration in a parallel processing environment. The research is in a developmental stage. Preliminary numerical results are presented for several example problems.

NOMENCLATURE

SS_{ijk}	- j^{th} subsystem at level i with parent k at level (i-1)
x^{ijk}	- vector of design variables for SS_{ijk}
y^{ijk}	- vector of design parameters for SS_{ijk}
$C^{ijk}(x^{ijk},y^{ijk})$	- cumulative constraint violation function for SS_{ijk}
$F^{ijk}(x^{ijk},y^{ijk})$	- penalty function for SS_{ijk}
$f^{ijk}(x^{ijk},y^{ijk})$	- objective function for SS_{ijk}
$g_w^{ijk}(x^{ijk},y^{ijk})$	- vector of inequality constraints for SS_{ijk}
$h_v^{ijk}(x^{ijk},y^{ijk})$	- vector of equality constraints for SS_{ijk}
KS	- Kresselmeir - Steinhauser function

INTRODUCTION

A modern vehicle (aircraft or automobile) is a complex engineering system composed of many subsystems that are tightly coupled. Often, the number of design variables involved and constraints imposed is large. The total amount of data handled becomes so large that synthesis of such a system is both intractable and costly and can easily saturate even the most advanced supercomputers available today. A remedy is to break the large problem into several manageably smaller subproblems; and solve these subproblems independently without loosing integrity with the main or parent problem and ultimately achieve satisfactory results for the original problem. This method of problem solving is known as decomposition.

Decomposition is the technique of dividing a large task into a set of smaller, self-contained subtasks that can be solved concurrently [1].

In a multilevel decomposition technique, the main structure at the top level is decomposed into a hierarchical tree consisting of substructures at different levels. A typical multilevel decomposition tree is shown in Figure 1. The main advantage of the multilevel optimization scheme lies in the fact that all substructures can be analyzed and optimized independently with convenient coupling. This application makes such an algorithm appropriate for parallel computing technology existing today.

Several studies have been devoted to the decomposition of large-scale optimization problems [2]. Mainly two classes of decomposition methods exist: ***formal methods*** and ***intuitive methods***. *Formal methods* decompose a problem using its mathematical structure. Such a decomposition may be fully automatic and can be built-in within the design cycle. However, in *intuitive methods*, an understanding of the physics of the system is the prime factor directing the decomposition. These intuitive methods provide an alternative for decomposing those problems which do not possess the structure for which a formal decomposition method exists.

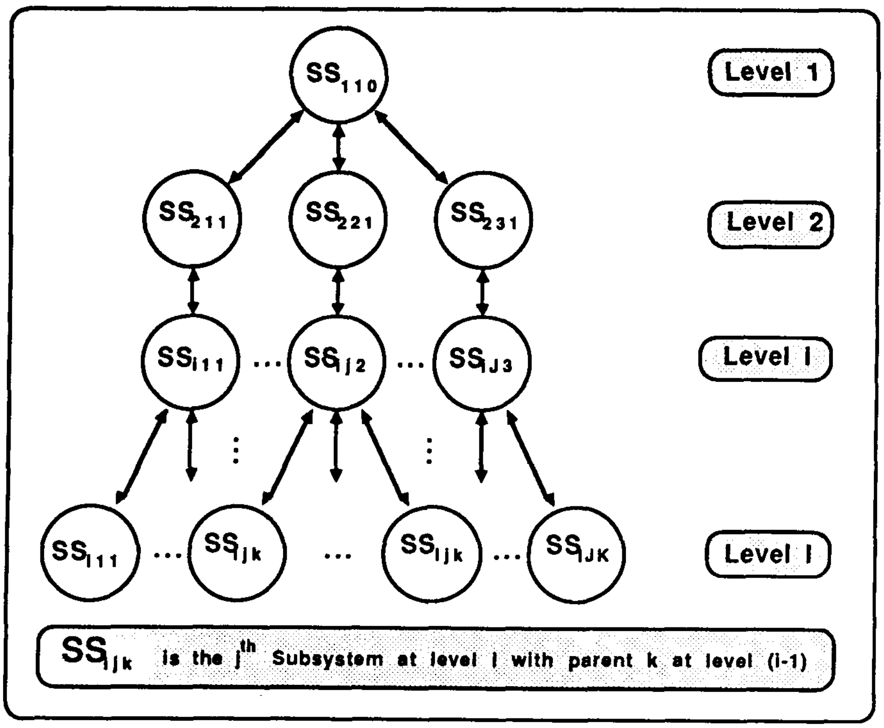

Figure 1
A general, Multilevel Decomposition Tree

PARALLEL PROCESSING

In a broader sense, the terms parallel computing or concurrent processing are used to define simultaneous execution of multiple tasks on multiple CPU's. This is achieved either by synchronizing the tasks on a multiprocessor or by effectively distributing the tasks among a network of computers. Parallel computing operations are classified in a number of ways depending upon the architecture of available computing resources and the granularity of the applications.

Flynn [3] has divided computer architectures from a macroscopic point of view using stream concept. Stream in this context simply means a sequence of items (instructions or data) as executed or operated on by a processor. The four broad classifications of machine organizations are:

(1) ***The Single Instruction stream - Single Data stream (SISD)***, which represents most conventional, uni-processor computing equipments available today.

(2) ***The Single Instruction stream - Multiple Data stream (SIMD)***, which includes most array processors; for example, Illiac IV.

(3) ***The Multiple Instruction stream - Single Data stream (MISD)*** type organizations; for example pipeline computers like CYBER 205.

(4) ***The Multiple Instruction stream - Multiple Data stream (MIMD)*** machines, which include the multiprocessor systems and distributed computing networks. It is possible to classify the MIMD architecture further according to coupling of the multiple processors as ***Tightly-Coupled*** and ***Closely-Coupled*** systems.

Various parallel computing applications are also classified based on granularity ranging from infinite grain size to very fine grain size. Granularity is measured by synchronization interval which is in fact the period between synchronization events measured in number of instructions for multiple processors or processing elements.

Concurrent processing computers set new demands on data structure, data management, organization, program coding, and adaptability considerations. These computers offer the possibility for significant gains in computational speed for structural synthesis based on multilevel optimization. Experience in parallel processing on NASA Langley's first multiple instruction, multiple data (MIMD) computer has shown that the greatest computational gains are obtained by writing special-purpose codes based on "rethinking" the solution method; somewhat smaller gains have resulted from "recoding" an existing algorithm, and no gain has resulted from the approach of just running an existing program on a parallel computer [4].

In recent years, significant research effort has been reported in applying linear and nonlinear finite element algorithms on concurrent processing computers using substructuring methods. Kowalik [5] has projected the potential for impact of parallel computers on numerical algorithms. Lootsma and Ragsdell [6] have described the state-of-the-art work in parallel nonlinear optimization area. A unique feature of this research is that this is the first attempt to implement a multilevel decomposition code on a network of computers. For this a VAX 8650 will be used as a host to load applications on a number of microvax stations available in a network of computers at the Design Productivity Center, University of Missouri, Columbia.

MULTILEVEL DECOMPOSITION

Most of the multilevel optimization algorithms developed so far involve intuitive or physical decomposition of the large-scale system into its component subsystems. All these algorithms exhibit a general philosophy of design. The subsystems are designed separately as component level synthesis problems. Then, the main system and all the subsystems are coupled appropriately and synthesized so as to achieve overall convergence of the system. A few approaches to multilevel decomposition are described now.

Review of Multilevel Decomposition Algorithms

Lucien Schmit Jr. and Ramanathan [7] introduced a multilevel approach to the design of minimum weight structures so as to include both local and global buckling of the elements and the system. In a two-level formulation of their decomposition algorithm, they treated total structural weight as the system level objective function whereas at component level, instead of component weight as an objective function, they considered minimization of change in equivalent system stiffness of the component. This is due to the fact that a structure made up of minimum weight components is not necessarily a minimum weight system. Schmit & Ramanathan observed that an efficient multilevel decomposition scheme should inherently lead to a weaker and weaker coupling between the subsystems as the iterations proceed.

Another interesting decomposition approach is by Uri Kirsch [8]. In this approach, the design quantities are divided into a set of dependent design variables and another set of independent quantities called behavior variables. The design variables and the behavior variables are optimized at different levels leading to a minimum two-level optimization algorithm. In the proposed scheme, the top level system is decomposed into a number of subsystems as second level problems. At the first level, dependent design variables are optimized for any assumed behavior (independent) variables. Then, at second level behavior quantities are optimized for each subsystem separately. The third level is an optional level in which elastic analysis is repeated only after a complete solution of both the first and the second levels. The main advantages of this algorithm are that the number of elastic analyses required is small, the first level is decomposable and the number of independent variables is not affected by the number of loading conditions the structure is subjected to.

One of the most irksome problems with the multilevel or hierarchical decomposition approach is the discontinuous behavior of derivatives that is transferred from the lower levels of the hierarchy to the upper levels. Raphael Haftka [9] has proposed a hierarchical algorithm that is free of such difficulties. In this algorithm, a penalty function method is employed in combination with Newton's method with approximate second derivatives to perform the optimization.

Jaroslaw Sobieski at NASA LangleyResearchCenter proposed an intuitive scheme of multilevel decomposition in 1982 which is not only versatile but also convenient. In the Sobieski approach, a large-scale system is physically decomposed into a number of subsystems at multiple levels with the complex system at the top level and the detailed most elements at the lower most level. For each subsystem the design space is divided into a set of constant design parameters and another set of design variables. A unique feature of the algorithm is Optimum Sensitivity Analysis (OSA). Sensitivity of the subsystems to problem parameters is determined and this information provides for the vertical coupling between a subsystem and its parent subsystem. In order to optimize a complete vehicle configuration, the Sobieski algorithm is selected as a vehicle for structural synthesis [10].

Optimum Sensitivity Analysis

In a multilevel decomposition algorithm it is essential to estimate the sensitivity of a problem at its optimum to the assumed constant parameters of the problem. This information provides for necessary coupling between various subsystems in the decomposition tree. The optimum sensitivity coefficients are essentially the Lagrange Multipliers. Numerically, they correspond to total derivatives of the objective function and the design variables with respect to the design parameters. A number of methods have been developed to compute the sensitivity coefficients directly. These methods include: Lagrange multiplier method, penalty function methods, feasible directions methods with the extension of the latter method to incorporate higher order coefficients and discontinuities.

In the Lagrange multiplier method [11], one starts with the Kuhn-Tucker conditions for a constrained minimum. Noting that the optimum value of the design variable is given by $x^* = x^*(y)$ where, y is the vector of design parameters, one can differentiate the Kuhn-Tucker conditions with respect to y, using the chain rule of partial differentiation. On simplification, we get a set of simultaneous linear equations with the sensitivity derivatives $\left(\frac{\partial x^*}{\partial y}\right)$ as unknowns.

In the penalty function methods, the penalty function: $F(x,y) = f(x,y) + rP$ is differentiated with respect to the design parameter y. Here, f is the objective function, r is the penalty parameter and P is the penalty term which could be either an interior function, or an exterior function or in case of coupled constraints, a Kresselmeir - Steinhauser function which is essentially the envelope of the constraint surface. Depending upon the choice of penalty term, different sets of linear equations can be developed to solve for the unknown sensitivity values. Optimum sensitivity analysis based on penalty function formulation is adopted in this research.

Vanderplaats [12] has developed algorithm based on the method of feasible directions to compute linear and higher order sensitivity coefficients.

The Sobieski Algorithm

The Sobieski algorithm for multilevel decomposition is depicted in an easy to comprehend flow chart as shown in Figure 2.

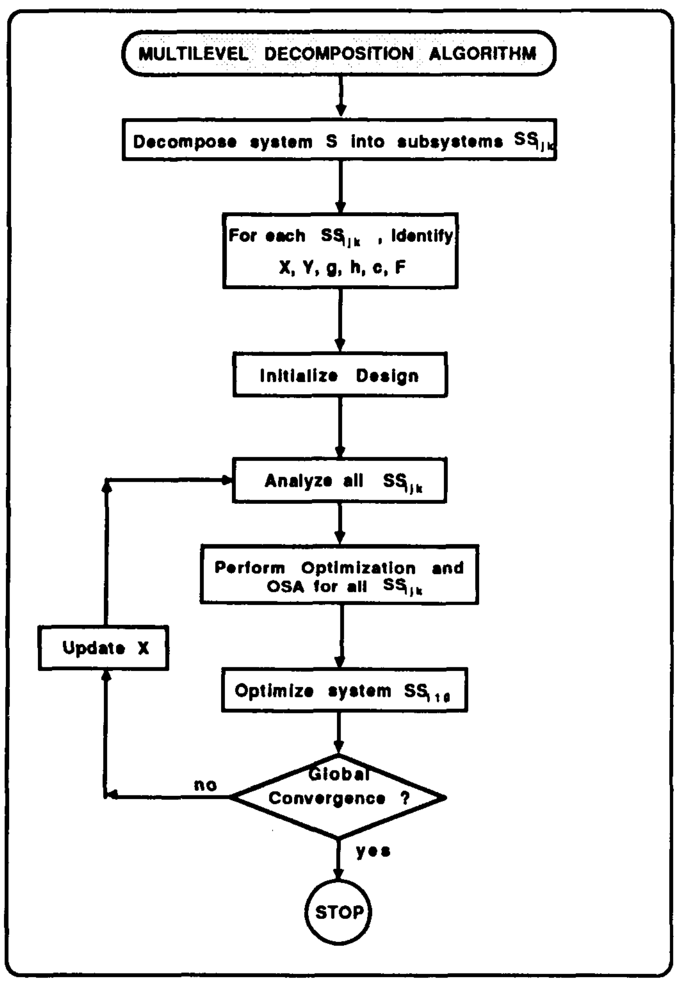

Figure 2
A Flow Chart for Multilevel Decomposition

The system $S=SS_{110}$ is decomposed into a number of subsystems SS_{ijk}. The design variables and the design parameters for the subsystems are x^{ijk} and y^{ijk} respectively. The design variables at the system level are essentially a set of design parameters for all the subsystems. The design procedure begins with initializing the design and analyzing all the subsystems. The objective functions for subsystem optimization are formulated by a cumulative constraint violation function C^{ijk} using, for example, a Kresselmeir-Steinhauser function of the form:

$$C^{ijk} = \frac{1}{p} \ln\left[\sum_{w} e^{-pg_w^{ijk}}\right] + p\sum_{v}\left(h_v^{ijk}\right)$$

During the system optimization, when the design variables x^{110} are perturbed, the changes in subsystem objective function and design variables can be predicted using linear the Taylor series extrapolation:

$$C_e^{ijk} = C^{ijk^{(*)}} + \sum\left(\frac{dC^{ijk^{(*)}}}{dx^{110}}\right)\left(\delta x^{110}\right)$$

$$x_e^{ijk} = x^{ijk^{(*)}} + \sum\left(\frac{dx^{ijk^{(*)}}}{dx^{110}}\right)\left(\delta x^{110}\right)$$

where, $\left\{\frac{dC^{ijk^{(*)}}}{dx^{110}}\right\}$ and $\left\{\frac{dx^{ijk^{(*)}}}{dx^{110}}\right\}$ are the total derivatives computed by Optimum Sensitivity Analysis. The procedure terminates when, (i) system response constraints are met, (ii) cumulative constraint violation for all the subsystems is reduced to at least zero and (iii) no further reduction of system mass appears possible.

EXAMPLES

The multilevel decomposition algorithm based on the Sobieski approach is being coded on VAX 8650 using FORTRAN 77 in double precision. In order to check the validity of the OSA algorithm and the multilevel decomposition algorithm various example problems have been set up. Closed-form analyses are generated for a three-bar truss problem and a portal frame involving beam elements. A highly simplified finite element model of an automobile configuration is also created using beam elements.

Three-Bar Truss Problem

The optimum sensitivity analysis is applied to a simple Three-Bar Truss (Figure 3), where the design variables are the areas of cross-section and the design parameters are the applied load P and the angle α which P makes with the y-axis. The weight of the structure is to be optimized with respect to the design variables. The behavior constraints correspond to simple upper and lower limits on stresses within the rods while the variable bounds ensure nonnegative areas of cross-section. Numerical results, shown in Figure 4, are in agreement with published results.

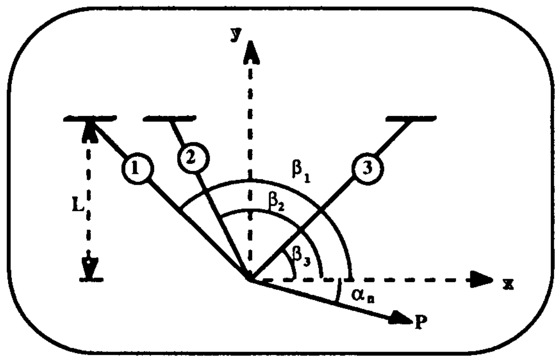

Figure 3
A Three-Bar Truss

Optimum Sensitivity Analysis

Optimum Design

$X_1^* = A_1^* = 0.7887$ (In^2)
$X_2^* = A_2^* = 0.4093$ (In^2)
$F^* = W^* = 2.64$ (Lbs)

Design Parameters

$Y_1 = P = 20000.$ (Lbs)
$Y_2 = \alpha = -45.$ (Degrees)

	Direct Differentiation	Penalty Function based Algorithm	Feasible Direction based Algorithm
$\left(\frac{dx_1^*}{dP}\right)$	3.940×10^{-5}	3.944×10^{-5}	4.428×10^{-5}
$\left(\frac{dx_2^*}{dP}\right)$	2.049×10^{-5}	2.0404×10^{-5}	6.711×10^{-6}
$\left(\frac{dF^*}{dP}\right)$	1.319×10^{-4}	1.319×10^{-4}	1.319×10^{-4}
$\left(\frac{dx_1^*}{d\alpha}\right)$	-0.8849	-0.885	-0.479
$\left(\frac{dx_2^*}{d\alpha}\right)$	0.9797	0.980	-0.169
$\left(\frac{dF^*}{d\alpha}\right)$	-1.524	-1.524	-1.524

Figure 4
Three-bar Truss Results

Portal Frame Problem

A portal frame with three beam elements is selected for testing the two-level decomposition algorithm. Figure 5 displays the complete frame as a top level system whereas, the individual beam elements are three subsystems for the problem. The portal frame structural weight is minimized subject to stress, deflection and buckling constraints. The simple statically indeterminate frame is analyzed using a variational method based on the minimization of the total complementary energy functional. Figure 6 shows the iteration history for "one-level" optimization versus "two-level" optimization.

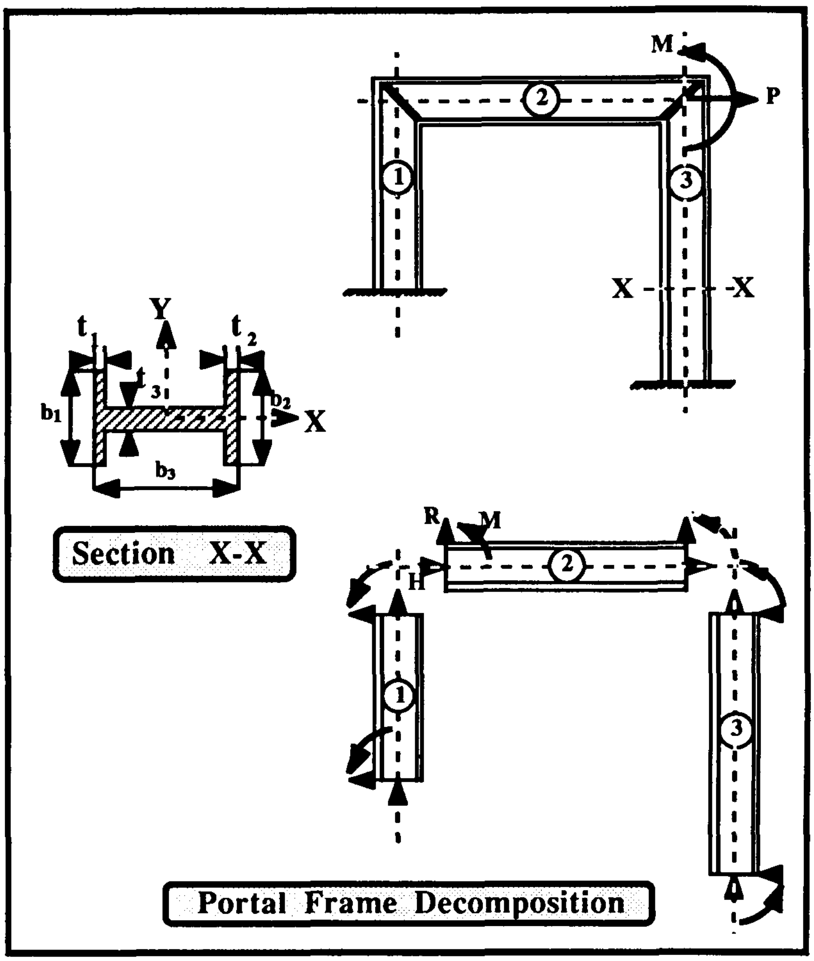

Figure 5
Portal Frame Problem

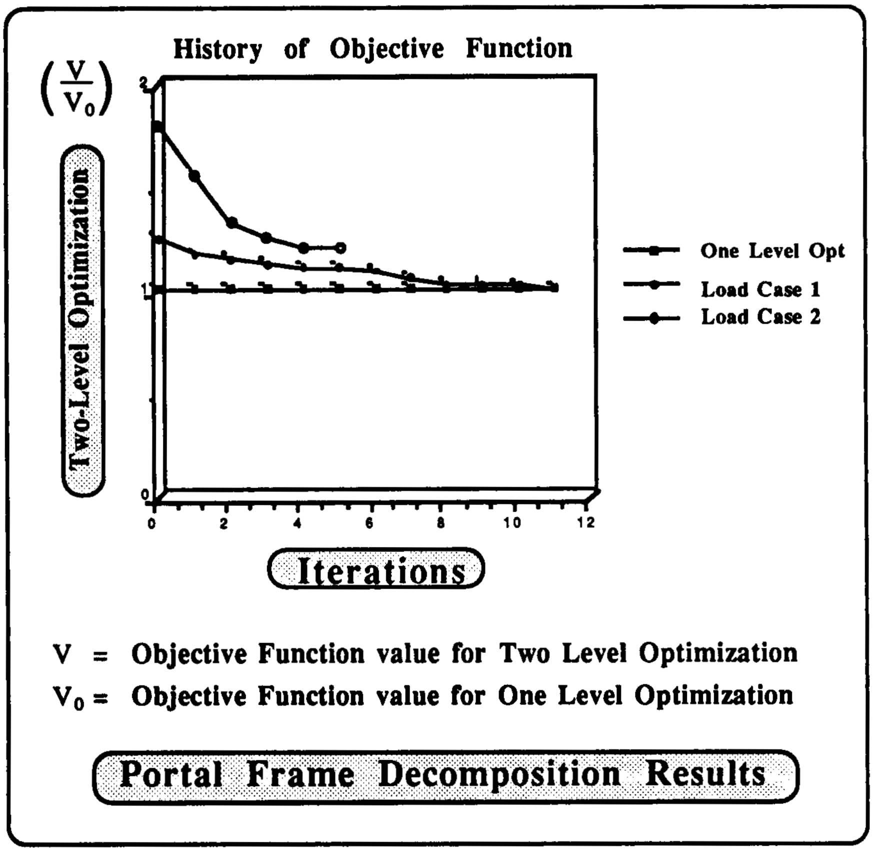

Figure 6
Portal Frame Decomposition Results

Optimum Vehicle Configuration Problem

A major objective of the study is to demonstrate the applicability of the multilevel decomposition algorithm to produce an optimum automobile configuration from a generic model of a complete automotive structure. As a first step in this direction, a simplified uniframe car model is developed [13]. Figure 7 shows the car frame selected to develop a simplified finite element model containing 22 nodes with 33 beam elements. The model is being studied for its static and dynamic response using NASTRAN. The automobile configuration is ideally suited for multilevel decomposition. In a three level decomposition strategy, the complete frame is selected as the top level which is decomposed into three middle level subsystems as shown in Figure 8. The third or bottom level corresponds to a box beam cross section.

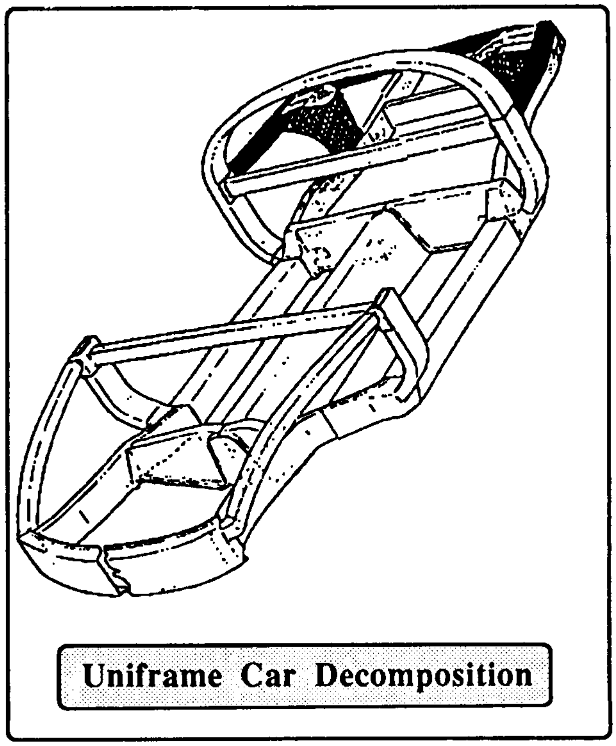

Figure 7
Uniframe Car Model

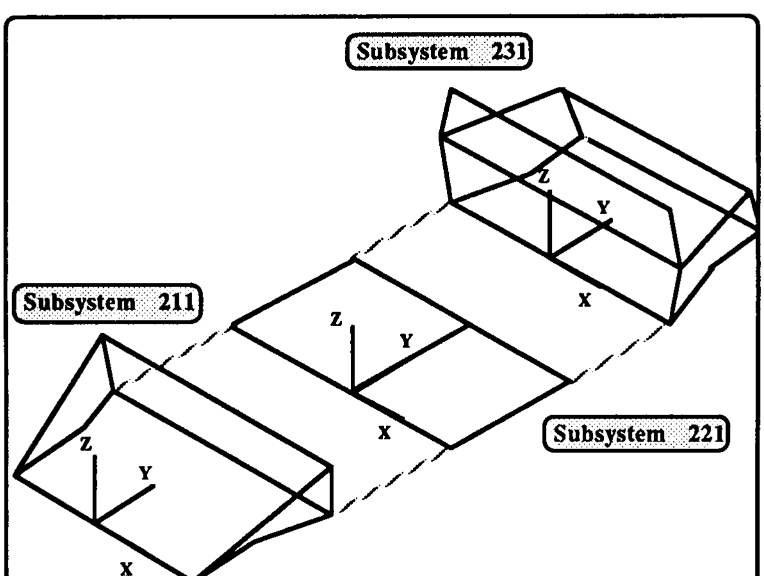

Figure 8
Second Level of Decomposition

CLOSURE

Multilevel decomposition is a multidisciplinary area. Here, we have concentrated on the study of large-scale structural synthesis in a parallel computing environment. A long term commitment to research in this area will have a significant impact on the design productivity of the automobile and aerospace industry of the future.

REFERENCES

1. Sobieski, J., "A Linear Decomposition Method for Large Optimization Problem - Blueprint for Development", NASA Technical Memorandum # 83248, Feb. 1982.

2. Barthelemy, J.,"Development of a Multilevel Optimization Approach to the Design of Modern Engineering Systems", NASA Contractor Report #172184, VPI, Virginia, Aug. 1983.

3. Flynn, M. J., "Some Computer Organizations and their Effectiveness", IEEE Transactions on Computers, Vol. C-21, No. 9, Sept. 1972.

4. Storaasli, O. and Bergan, Pal, "Nonlinear Substructuring Method for Concurrent Processing Computers", AIAA Journal, Vol. 25, No. 6, June 1987.

5. Kowalik, J. S., "Parallel Computers and their Impact on Numerical Algorithms and Related Software", Nonlinear Optimization 1981, NATO conference series, Academic Press, 1982.

6. Lootsma, F. A. and K. M. Ragsdell, "State-of-the-Art in Parallel Nonlinear Optimization", Design Productivity Center Report #86DPC027, University of Missouri, Columbia, Nov. 1986.

7. Schmit, L. A. and Ramanathan, R. K., "Multilevel Approach to Minimum Weight Design including Buckling Constraints", AIAA Journal, Vol. 16, No. 2, Feb. 1978.

8. Kirsch, U., "An Improved Multilevel Structural Synthesis Method", Journal of Structural Mechanics, #13(2), 1985.

9. Haftka, R., "An Improved Computational Approach for Multilevel Optimum Design", Journal of Structural Mechanics, 12(2), pp. 245 - 261, 1984.

10. Bhatt, V., Linden, S. and Ragsdell, K., "Optimal Synthesis of Large-Scale Systems using Decomposition Techniques and Parallel Processing", Design Productivity

Center Technical Report # 88DPC002, University of Missouri, Columbia, Feb. 1988.

11. Sobieski, J., J. Barthelemy and K. Riley, "Sensitivity of Optimum Solutions of Problem Parameters", AIAA Journal, Vol. 20, No. 9, Sept. 1982.

12. Vanderplaats, G., "Efficient Calculation of Optimum Design Sensitivity", AIAA Journal, Vol. 23, No. 11, Nov. 1985.

13. Linden, S., "Decomposition and Multilevel Optimization of Large Scale Systems", Final Report, Design Productivity Center, Columbia, May 1987.